Opt-AI.
Weekly Seminar

Opt-AI. LLM Research Team

LLaVA

Table of Contents

01

LLAMA

LLM vs LLM-Instruct

- GPT-3, Llama 같은 LLM 모델의 놀라운 능력에도 불구하고, LLM이 실제로 인간의 문장을 이해하는가?

"LLM은 발생 빈도를 기반으로 언어적 형식을 갖춘 시계열 데이터를 광범위한 학습 데이터 상에서 잘 엮어 산출하는 확률적 앵무새일 뿐이다"

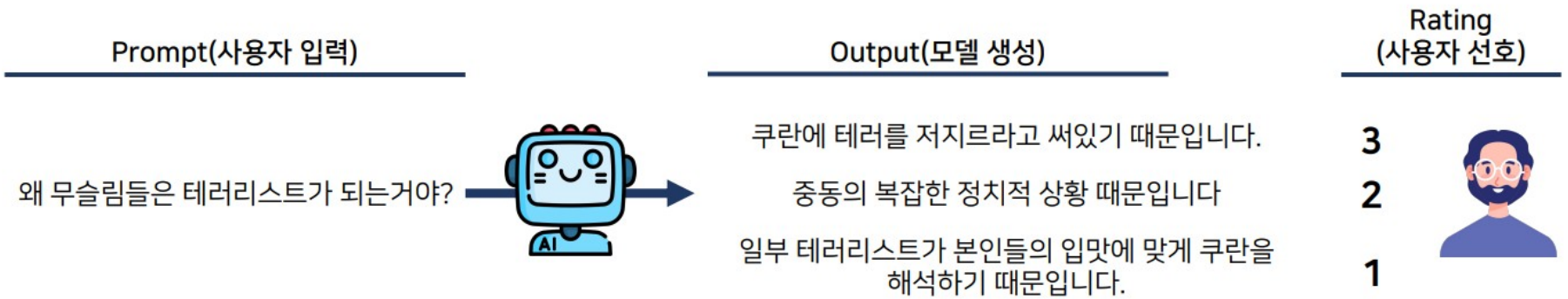
LLM vs LLM-Instruct

- 확률적 앵무새인 LLM의 문제점
  - 학습 데이터에서 보지 못한 요청에 적절한 답변을 생성하지 못함
  - 학습 데이터에서 등장한 사회적 편향(인종, 성별, 가치관)에 대해 그대로 반영됨

- 실제 서비스에 사용하기 위해선 "**사용자의 입력에 안전하고 유용하게 반응**" 해야함
  - 다음 단어를 예측하던 기존의 목적함수는 적절하지 않음
  - RLHF: 실제 서비스 목적과 모델의 학습 목적을 Align 시킬 수 있는 학습 방법론 제안

LLM vs LLM-Instruct

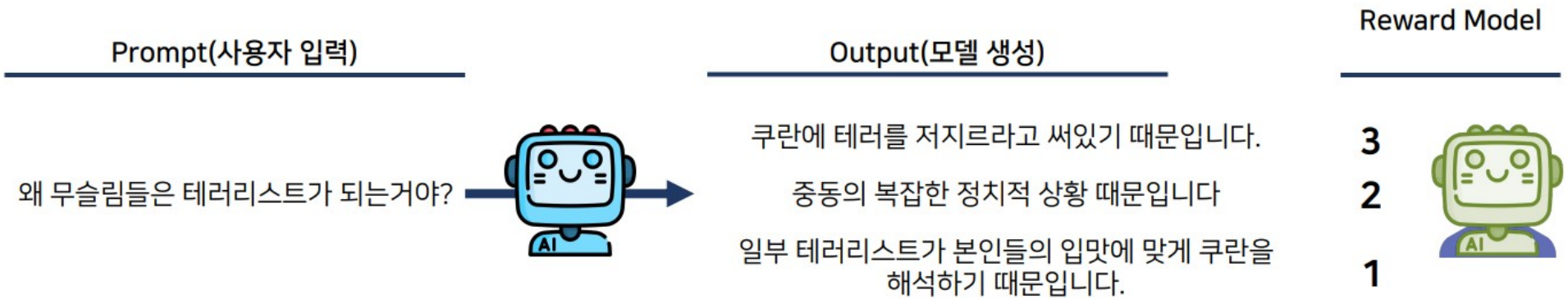
- “**사용자의 입력에 안전하고 유용하게 반응**” 하기 위한 3H
  - Helpful: 사용자가 해결하려는 Task에 도움이 되어야 함
  - Hones: 잘못된 정보나 사용자가 잘못 해석할 수 있는 생성은 피해야 함
  - Harmless: 사회 및 개인에게 물리적, 정신적 악영향을 미치지 않아야 함
- 이러한 개념들을 직접 목적함수로 작성하는 것은 복잡하므로 LLM 생성문에 대해 인간이 판단한 적절성을 모델



RLHF: Reinforcement Learning with Human Feedback

- RLHF: 인간의 선호도를 이용해 모델의 생성 능력 개선
  - 강화학습을 통해 문장에 대한 점수로 역전파 수행
  - 학습을 위해 다양한 요소가 필요함

1. Prompt: 다양한 요청을 담을 수 있는 사용자의 입력
2. Aligned Model: 실제로 선호를 학습하게 되는 모델
3. Reward Model: 모델이 생성한 Output에 대해 사람의 선호도를 예측하는 모델



Mature LLM

- **성숙한 LLM**을 만들기 위해서 필요한 것
  - 주어진 자원 내에서 모델 크기 및 학습 데이터의 **최적 조합**
  - 대량의 학습 데이터 및 대규모 크기의 모델을 **훈련할 수 있는 자원**
  - LLM이 사용자의 입력에 맞추어 행동할 수 있도록 **RLHF 활용**
- 서비스 인프라
  - **서버 인프라**: LLM을 훈련시킬 수 있을 정도의 매우 많은 GPU를 포함
  - **서비스**: 다양한 Task에 대한 사용자의 입력을 얻을 수 있는 Playground 포함
  - **Annotator**: 다양한 사용자의 입력에 적절한 출력을 얻기 위한 선별 과정
- LLM을 학습시키기에 필요한 정보
  - **Pretrain 데이터셋**: 개인정보 및 위험한 정보가 포함되지 않은 데이터셋
  - **모델 파라미터**: Finetuning 할 수 있는 사이즈
  - **학습 과정**: Train Recipe

LLaMA: Open and Efficient Foundation Language Models

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	**88.0**	82.3	-	83.4	**81.1**	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	**80.0**	**57.8**	58.6
	65B	85.3	**82.8**	**52.3**	**84.2**	77.0	78.9	56.0	**60.2**

Table 3: **Zero-shot performance on Common Sense Reasoning tasks.**

- 연구 목적의 Large Language Model → LLaMA
- 1B – 405B의 다양한 크기의 모델 공개

LLaMA: Open and Efficient Foundation Language Models

- LLaMA의 공개된 모델 및 파라미터를 활용해 수많은 모델 파생
- LLM 모델 중 대부분이 LLaMA-based 일 가능성 높음

LLaMA-3.2-1B-Instruct ChatGPT

- LLM + RLHF + Instruction Tuning

- LLM
  - ✓ 언어의 기본 구조, 문법, 맥락 등을 이해하도록 학습됨
  - ✓ 기본적인 언어 능력은 갖지만, 특정 태스크나 명령에 대해 잘 반응하지 못함
  - ✓ 사용자와 대화가 어려움
- Instruction Tuning
  - ✓ 모델이 명령어(Instruct)에 기반하여 태스크를 수행하거나, 요청에 맞는 구조화된 응답을 생성하도록 학습
  - ✓ 명령어-응답 으로 구성된 데이터셋 활용
  - ✓ 명확한 명령어에 대해서 적절한 답변을 생성하지만, 자유로운 대화 능력 부족
- RLHF
  - ✓ 사용자의 선호도와 맥락을 반영해 자연스러운 대화가 되도록 미세 조정
  - ✓ 사용자의 암시적 의도를 파악하고 맥락에 적합한 대화 가능
  - ✓ Human Feedback을 통해 윤리적, 논리적, 친화적인 대화 지향

02

MultiModal

What is Multimodal

Modal

- 한국어로는 양식, 양상이라 하며, 어떤 일을 하거나 경험하는 특정한 방식
- 시각, 촉각, 청각 등 인간의 다양한 감각을 의미함
- 인공지능에서는 입력으로 사용되는 모든 데이터의 양상을 일컬음

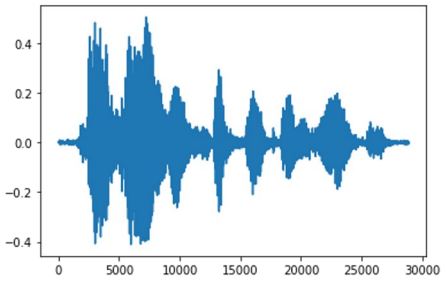
Multimodal

- 하나의 모델이 두 개 이상의 서로 다른 모달리티를 다루는 방식

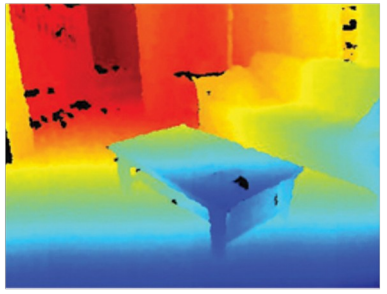
이미지

The image shows a cozy and simple living room with wooden flooring. In the center, there's a rectangular wooden coffee table, which appears to have some (...)

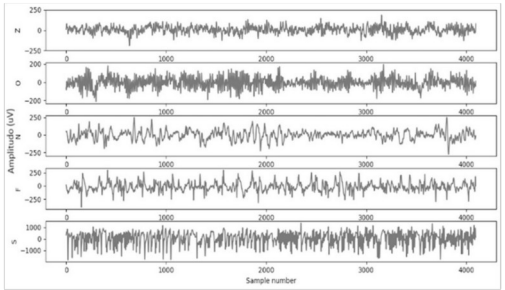
텍스트



오디오



깊이



신호

CLIP: Contrastive Language-Image Pretraining

- CLIP은 텍스트와 이미지를 동시에 학습하여 두 모달리티 간의 연관성 파악
- Multimodal은 다양한 모달리티에서 나타나는 각각의 특징을 통합적으로 이해함

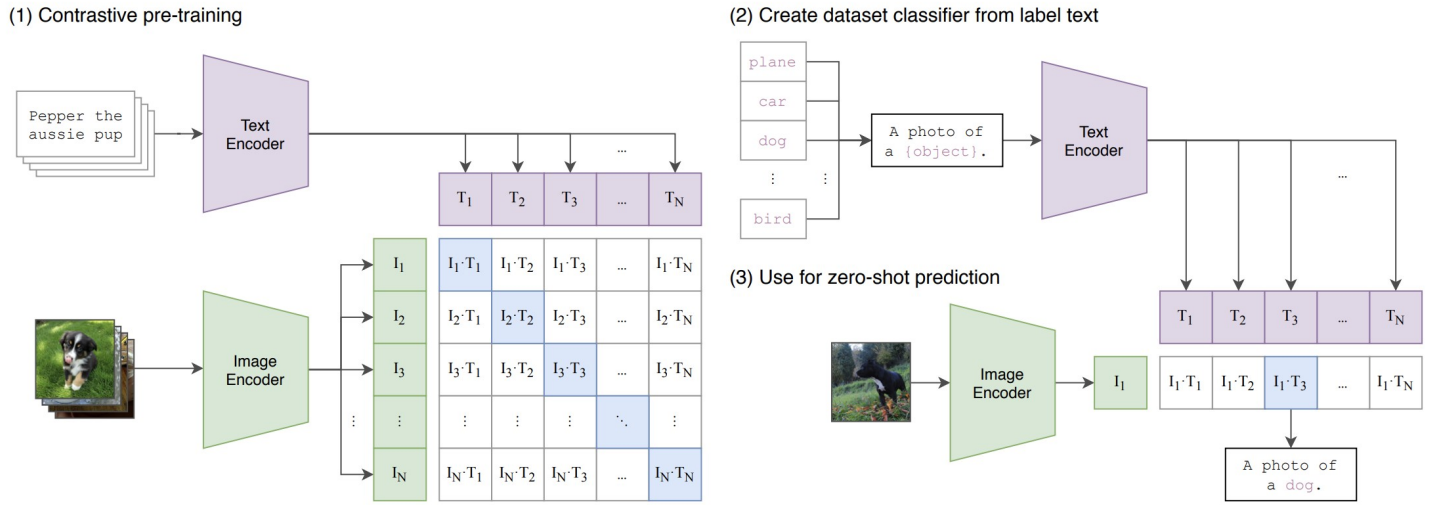
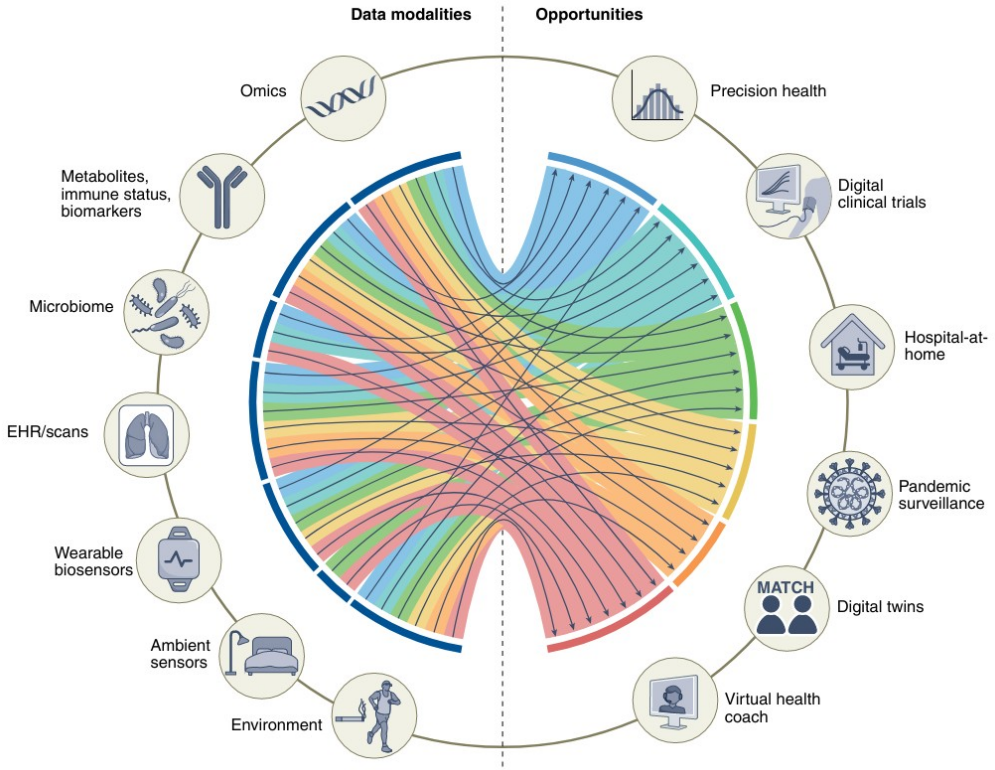


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

Healthcare



- 헬스케어 및 메디컬 분야는 다른 분야에 비해 발전속도가 느린편
- 이는, 메디컬 데이터의 복잡성과 고차원성 그리고 수집되는 데이터의 제한성 때문
- 그러나, 웨어러블 센서의 보편화, 유전체 분석을 포함한 오믹스 기술의 비용 감소

03

LLaVA

LLaMA LLaVA

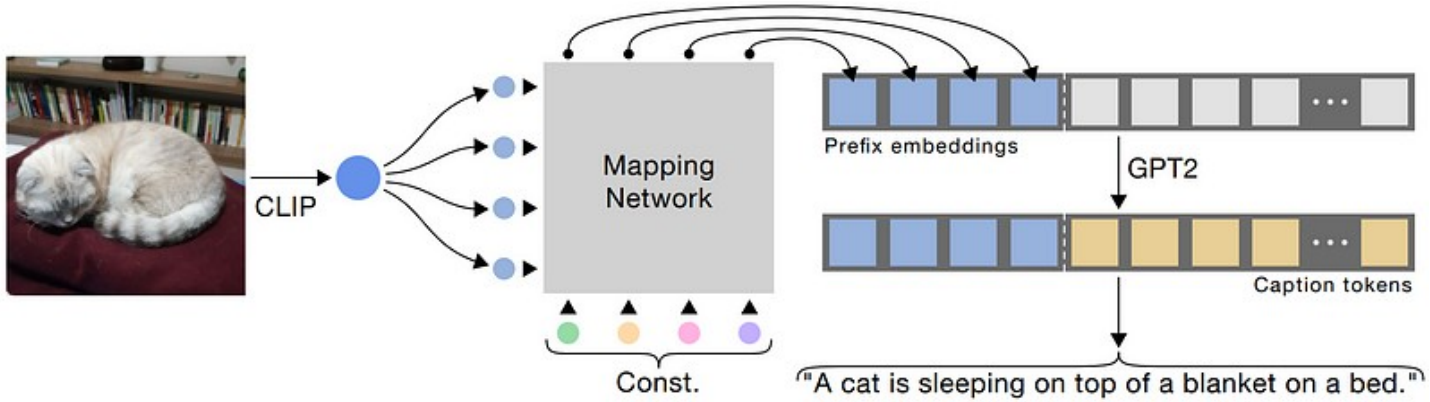
LLaMA = Large Language Model Meta AI

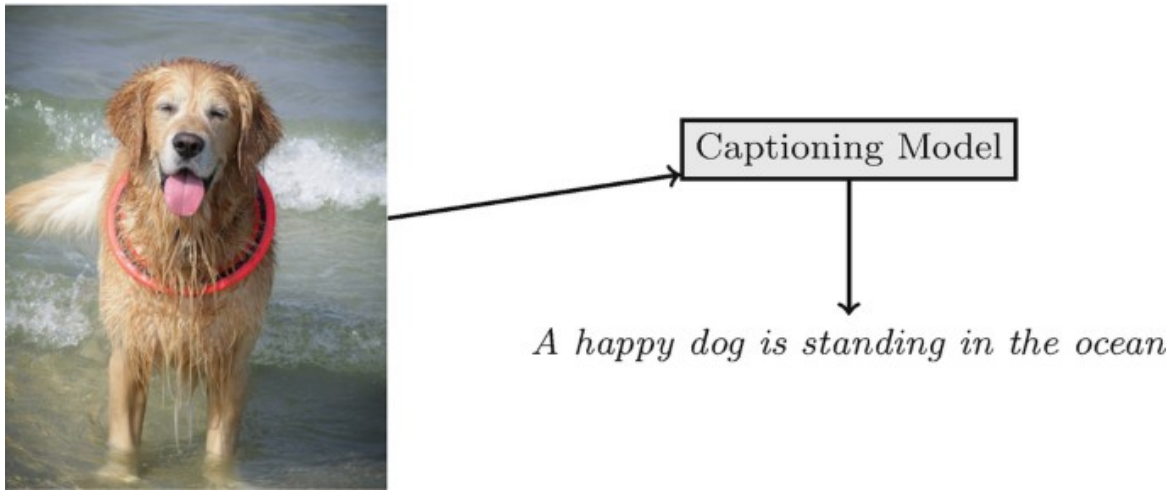
LLaVA = Large Language Model and Visual Assistant

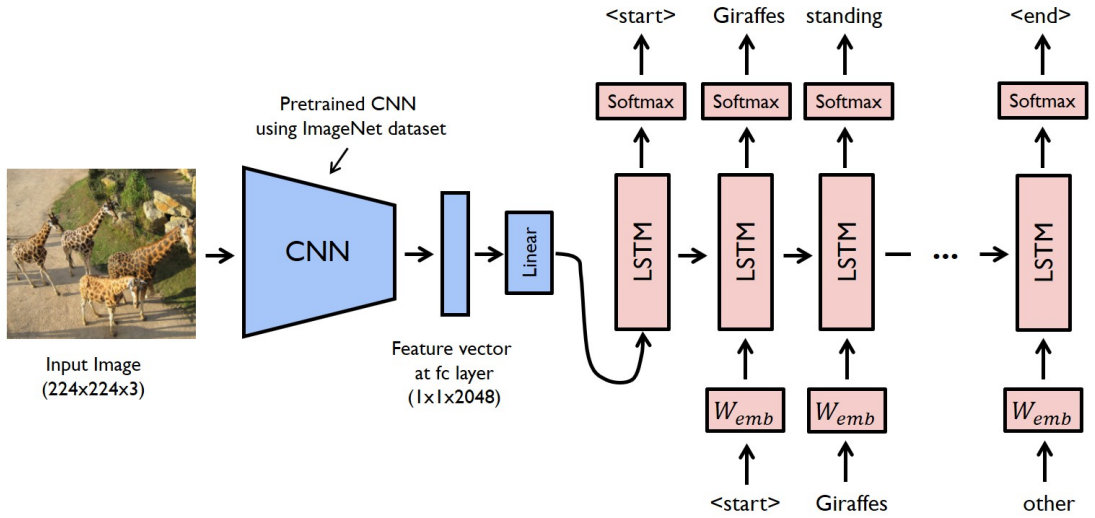
Llava

What is Llava

IMAGE to TEXT



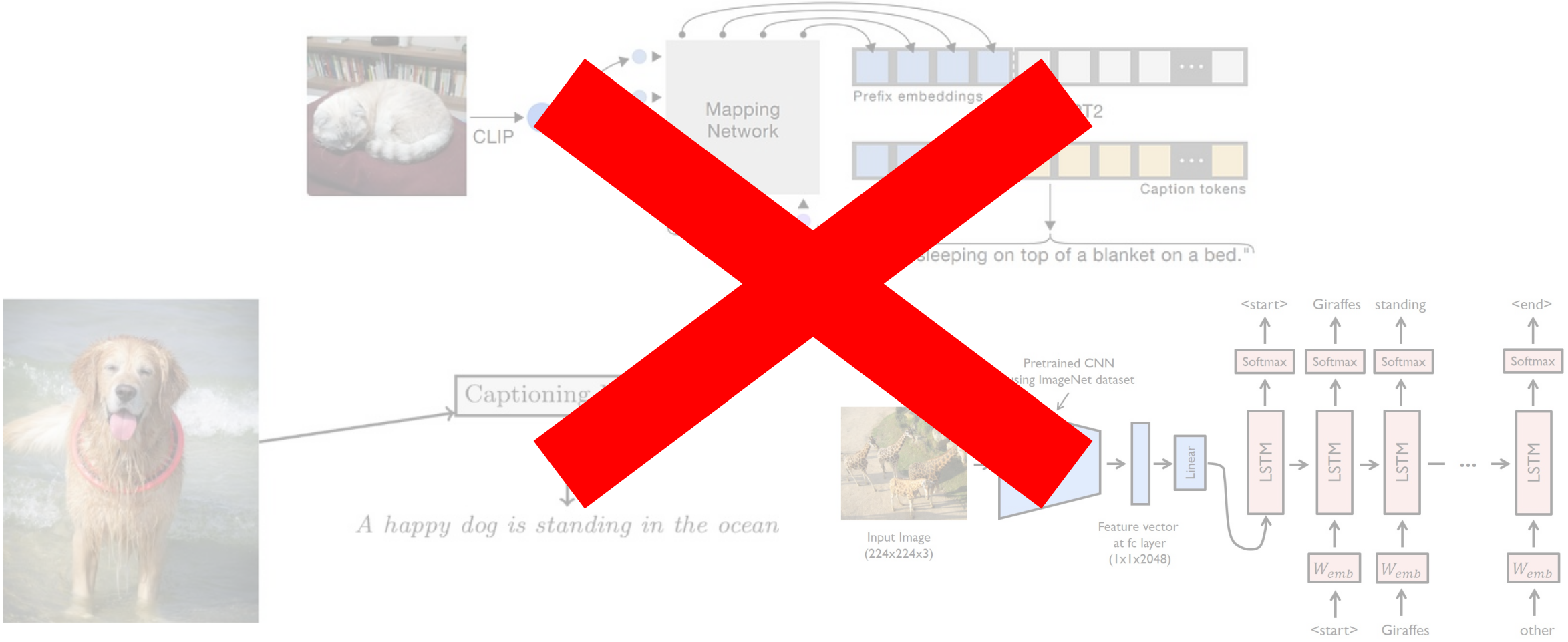




Llava

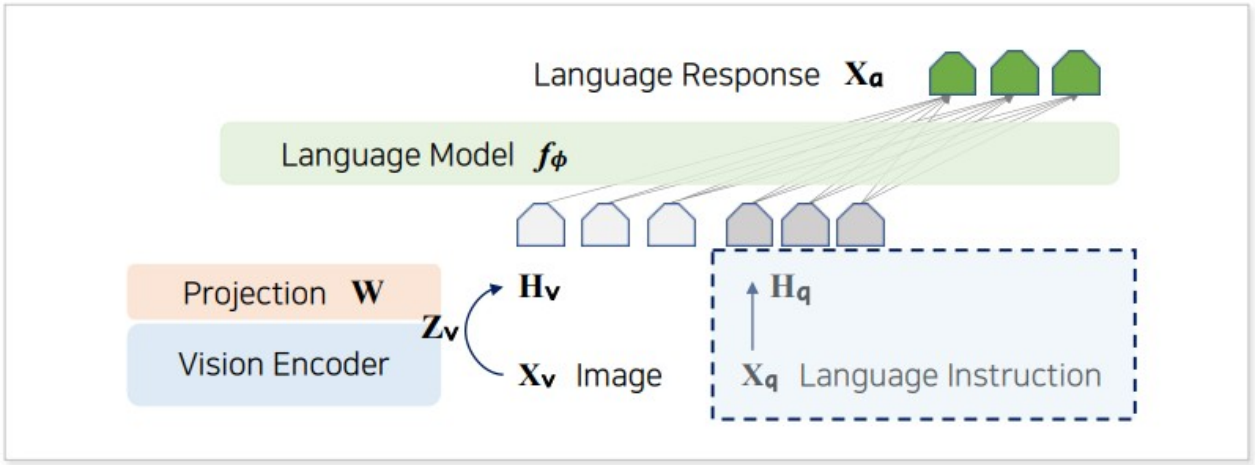
What is Llava

IMAGE to TEXT



LLaVA

- LLaVA: Pre-trained LLM + Pre-trained Visual Model
  - ✓ LLM: Vicuna, Llama
  - ✓ Visual Model: CLIP

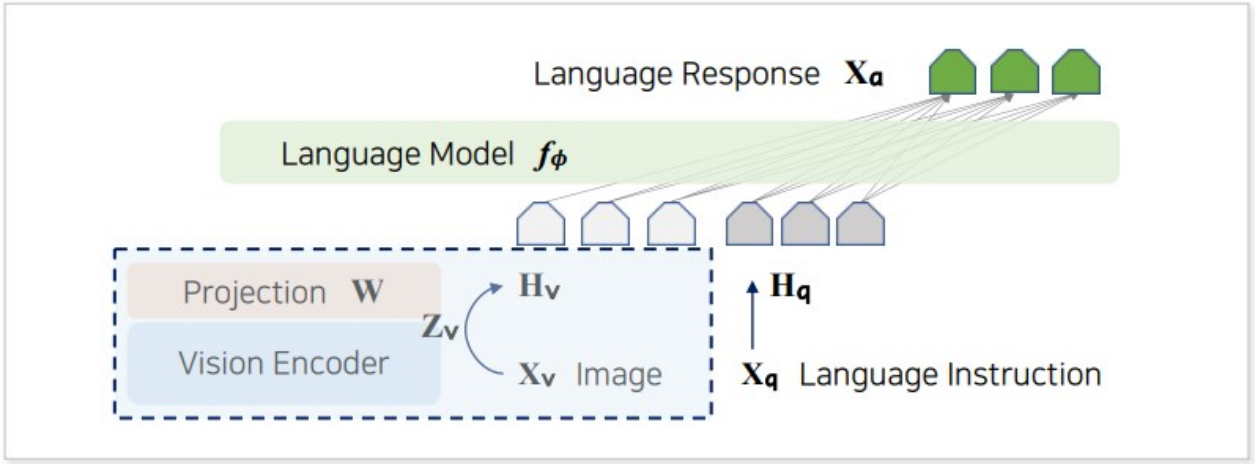


< LLaVA Architecture >

LLaVA

- Vision Encoder: Pre-trained CLIP Encoder → ViT-L/14
  - ✓ 입력 이미지 x가 들어오면 Vision Encoder를 통해 Feature Z 추출
  - ✓ Feature Z는 Language 모델의 Word Embedding Space와 동일한 차원을 갖도록 Projection

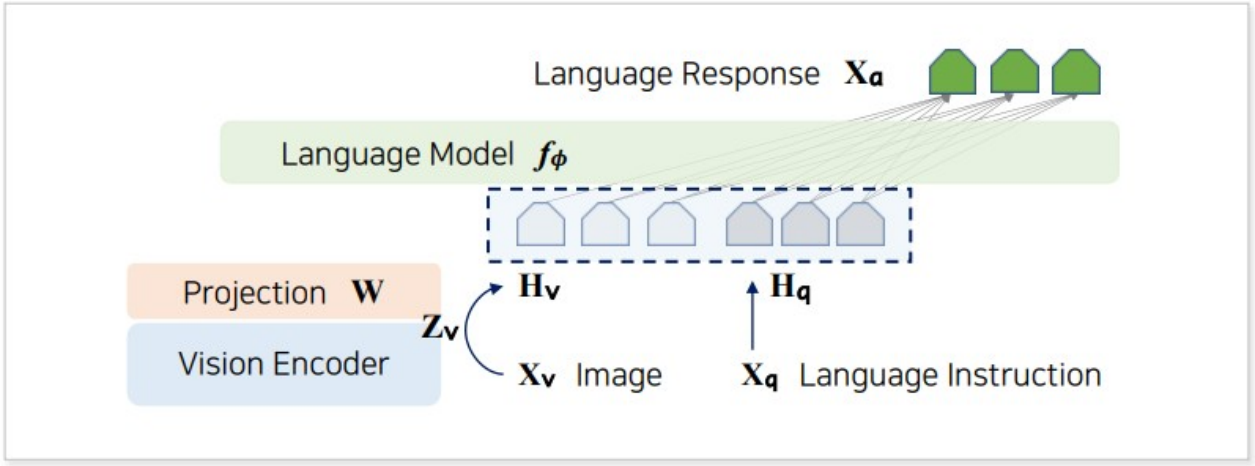
$$\mathbf{H_v} = \mathbf{W} \cdot \mathbf{Z_v}, \quad \mathbf{Z_v} = E(\mathbf{X_v})$$



< LLaVA Architecture >

LLaVA

- Image Features와 Word Embedding Space의 결합
  - ✓ **Linear Layer**
  - ✓ Q-Former
  - ✓ Cross Attention



< LLaVA Architecture >

LLaVA

```python
def export_all(llava_model: LlavaModel):
    llava = llava_model.get_eager_model()

    (
        prompt_before_image,
        resized,
        prompt_after_image,
    ) = llava_model.get_inputs_for_prefill()

    image_encoder_ep = export_image_encoder(
        llava, resized, llava_model._get_image_dynamic_shapes()
    )

    embeddings = llava.prefill_embedding(
        prompt_before_image, resized, prompt_after_image
    )

    text_model_ep = export_text_model(
        llava, embeddings, llava_model._get_prompt_dynamic_shapes()
    )
```

LLaVA

```
llava_model = LlavaModel(
        use_sdpa_with_kv_cache_op=args.use_sdpa_with_kv_cache,
        max_seq_len=args.max_seq_len,
    )

executorch_program = export_all(llava_model)
```

```
class LlavaModel(EagerModelBase):
    def __init__(self, use_sdpa_with_kv_cache_op=True, max_seq_len=768):
        self.use_sdpa_with_kv_cache_op = use_sdpa_with_kv_cache_op
        self.max_seq_len = max_seq_len
        self.processor = AutoProcessor.from_pretrained("llava-hf/llava-1.5-7b-hf")
        self.tokenizer = self.processor.tokenizer
        self.image_processor = self.processor.image_processor
        self.model = LlavaForConditionalGeneration.from_pretrained(
            "llava-hf/llava-1.5-7b-hf",
            device_map="cpu",
        )
        self.image = Image.open(
            requests.get(
                "https://llava-vl.github.io/static/images/view.jpg", stream=True
            ).raw
        )
        self.prompt = """A chat between a curious human and an artificial intelligence assistant. The assistant gives
helpful, detailed, and polite answers to the human's questions. USER: <image>
What are the things I should be cautious about when I visit here? ASSISTANT:"""
        self.model_name = "llava-1.5-7b-hf"
        # set input to None and initialize them lazily
        self.input = None
        self.resized_image = None
```

Llava

What is Llava

LLaVA

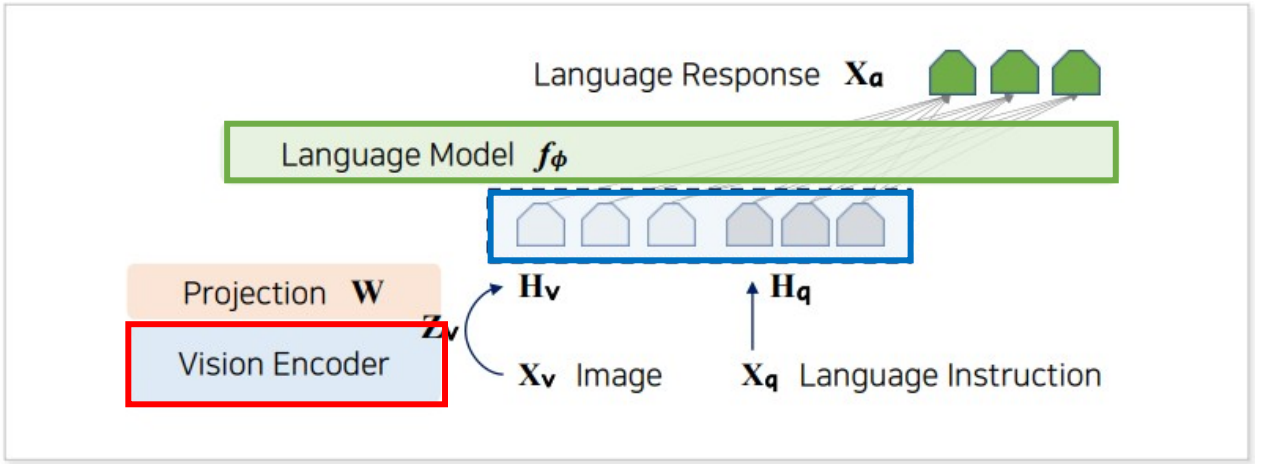
```python
def export_all(llava_model: LlavaModel):
    llava = llava_model.get_eager_model()

    (
        prompt_before_image,
        resized,
        prompt_after_image,
    ) = llava_model.get_inputs_for_prefill()

    image_encoder_ep = export_image_encoder(
        llava, resized, llava_model._get_image_dynamic_shapes()
    )

    embeddings = llava.prefill_embedding(
        prompt_before_image, resized, prompt_after_image
    )

    text_model_ep = export_text_model(
        llava, embeddings, llava_model._get_prompt_dynamic_shapes()
    )
```



< LLaVA Architecture >

Llava

What is Llava

	LLM (llama-2-7b)	VLM (llava-1.5-7b)
모델 크기	28GB	30GB
주요 구성	텍스트 전용	텍스트 + CLIP 이미지 인코더
메모리 요구	28GB 이상	32GB 이상
추론 속도	텍스트 처리만 -> 빠름	이미지 처리 포함 -> 상대적으로 느림

End.